Evolving to Recognize High-dimensional Relationships in Data: GA Operators and Representation Designed Expressly for Community Detection

Kenneth Smith
University of Missouri-St. Louis
St. Louis, MO 63121, U.S.A.
kpsc59@mail.umsl.edu

Cezary Janikow
University of Missouri-St. Louis
St. Louis, MO 63121, U.S.A.
janikowc@umsl.edu

Sharlee Climer
University of Missouri-St. Louis
St. Louis, MO 63121, U.S.A.
climer@umsl.edu

ABSTRACT

We present a new algorithm for network clustering based upon genetic algorithm methods to optimize modularity. The algorithm proposes an innovative, more abstract representation, along with newly designed domain-specific genetic operators. We then analyze the performance of the algorithm using popular real-world data sets taken from multiple domains. The analysis demonstrates that our algorithm consistently finds high quality or even optimal solutions without any *a priori* knowledge of the network or the desired number of clusters. Furthermore, we compare our results with five previously published methods and yield the highest quality for the largest of the benchmark datasets.

CCS CONCEPTS

• Theory of computation~Network optimization • Theory of computation~Evolutionary algorithms • Computing methodologies~Genetic algorithms

KEYWORDS

Community detection, clustering, genetic algorithm, modularity

1. Introduction

Networks, which are commonly used to model data sets, use a node to represent each object and an edge to represent a pair-wise relationship between two nodes. While it is often straight forward to calculate all pair-wise relationships, higher-order relationships are typically needed to provide actionable knowledge. Unfortunately, direct computations of all higher-ordered combinations are difficult for all but the smallest of datasets.

Exact optimization of a clustering objective is not feasible for large data sets and approximation techniques are needed [3]. Accordingly, a wide range of heuristic algorithms (e.g. [1, 3]), including genetic algorithm (GA) approaches have been proposed to solve this problem, (e.g. [2, 5–7, 9]). A majority of the GAs presented in the literature use one of two objective functions: modularity [2, 7] and clustering coefficient [9].

The algorithm presented in this manuscript is based on modularity as presented by [8]. The definition follows:

$$Q = \sum_i \left(\frac{l_i}{m} - \left(\frac{ds_i}{2m}\right)^2 \right) \qquad (1)$$

Here l_i is the number of intra-cluster edges in cluster i, ds_i is the sum of the degree of all nodes in cluster *i*, and *m* is the total number of edges in the original graph.

2. Algorithm Details

We propose a novel chromosome representation where each cluster is represented by an element in a primary linked list. A given cluster C_i then has its member nodes listed in a secondary list, which can be of different length for each cluster. This representation yields two primary benefits. First, intelligent problem-specific operators can be implemented efficiently and effectively. Second, when determining all of the nodes in a single cluster, only n_i values, the number of nodes in cluster C_i, must be checked instead of all nodes, as in the stand representation. We will call this new algorithm the linked-lists and multi-operator approach (LLAMA).

We utilize safe initialization [4], which ensures that each initial chromosome is constructed of connected clusters. A connected cluster is one in which there is a path from any node to any other node in the cluster. Half of the population is created using a Random Walk algorithm similar to [5]. The other half of the chromosomes are created by repeatedly splitting apart the original network, until a randomly chosen number of clusters is reached.

Listed below are the four mutation and two crossover operators. Both crossover operators require two chromosomes which will be identified as Chromosome 1 and Chromosome 2. Additionally, two nodes are neighbors if they share and edge in the original network. Before mutation or crossover occurs, tournament selection is used to identify the parent chromosomes and elitism guarantees the fittest individual is maintained.

Split. The Split operator partitions the nodes from a single cluster into two clusters, where each cluster is connected.

GECCO '19 Companion, July 13–17, 2019, Prague, Czech Republic
© 2019 Copyright is held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 978-1-4503-6748-6/19/07…$15.00
https://doi.org/10.1145/3319619.3322073

Merge. The Merge operator is used to combine nodes from two clusters into a single cluster. Two clusters are merged only if the resulting cluster is connected.

Redistribution. Two clusters are merged and then the resulting cluster is split.

Point Mutation. The node is moved to the same cluster as one of its neighbors or to a new cluster, as a singleton.

Node Crossover. In Chromosome 1, the node to crossover is moved to the same cluster as one of its neighbors, based on how the node is clustered in Chromosome 2.

Cluster Crossover. A cluster is randomly chosen in Chromosome 2. Each of the nodes in this cluster are located in Chromosome 1 and combined into a new cluster in Chromosome 1.

BFS Fix. Since some of the operators can create unconnected clusters, a Breadth First Search (BFS) is performed after the other operators. If any unconnected clusters are identified, the unconnected parts are moved to create new clusters.

3. Experimental Results

Our algorithm was evaluated 50 independent times for each of the networks listed below. Each trial was run for 5000 generation, with a population size of 100 and a tournament size of 8. The highest modularity after each trial was recorded and used to calculate a maximum, average, and standard deviation for LLAMA, as listed in Table 1. The average modularity for each algorithm, as reported in the literature, and details on each network are also included.

We examine the consistency of LLAMA using the standard deviation in modularity. The same solution was achieved on each trial for the *Karate* and *Polbooks* networks, while the largest range of solutions coincided with the largest network, *Email*.

Details on each algorithm can be found at: TGA [11], GAOM [6], ECD [2], GACD [10], and GALS [5]. In addition to the GALS results, [5] also provides a concise summary of the results from TGA and GACD. LLAMA demonstrated average modularity values that are similar to these leading methods and in fact outperformed all of the methods on the *Email* network.

4. Conclusion and Future Work

In this paper, we introduced a modularity-based network clustering approach using original GA methods that are designed specifically for this domain. A novel chromosome representation was introduced, which uses a primary linked-list to identify the clusters, and a secondary linked-list emanating from each cluster, that contains all the nodes within the cluster. Additionally, several GA operators are introduced that efficiently utilize this new chromosomal representation.

We compare the performance of LLAMA against five previously published GA approaches. Our algorithm is able to repeatedly find optimal, or near optimal solutions, without any *a priori* information. LLAMA was comparable to the other methods, and in fact out performed all the other GA methods for the largest data set, *Email*. This suggests a potential for this approach to provide accurate clustering results for increasingly large datasets. LLAMA populations can easily be run in parallel, and given adequate computational resources, it holds promise to scale up to massive datasets of interest in scientific, business, and government applications. In the future, we would like to add other initializations using some fast heuristic methods, and perform detailed analysis of usefulness of various operators introduced here.

Table 1: Results of LLAMA vs. Other GAs for Multiple Networks

Network	Nodes	Edges	Maximum	Std. Dev.	TGA	GAOM	EDC	GACD	GALS	LLAMA
Karate	34	78	0.4198	0.0000	0.4039	0.4183	0.38	0.4198	0.4198	0.4198
Dolphins	62	159	0.5285	0.0008	0.5241	---	0.46	---	---	0.5281
Polbooks	105	441	0.5272	0.0000	0.5245	0.5264	0.52	0.5272	0.5272	0.5272
Football	115	613	0.6046	0.0001	0.5937	0.6046	0.56	0.6044	0.6046	0.6045
Jazz	198	2742	0.4449	0.0003	0.4406	---	---	0.4435	0.4449	0.4444
Email	1133	5451	0.5756	0.0065	0.1871	---	---	0.4422	0.5599	0.5632

Acknowledgments: NIH 1RF1AG053303-01

References

[1] Aloise, D. and Caporossi, G. 2012. Modularity maximization in networks by variable neighborhood search. *10th DIMACS Implementation Challenge - Graph Partitioning and Graph Clustering*. (2012).

[2] Bilal, S. and Abdelouahab, M. 2017. Evolutionary algorithm and modularity for detecting communities in networks. *Physica A: Statistical Mechanics and its Applications*. 473, (2017), 89–96. DOI:https://doi.org/10.1016/j.physa.2017.01.018.

[3] Blondel, V.D. et al. 2008. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*. 2008, 10 (2008), 1–12. DOI:https://doi.org/10.1088/1742-5468/2008/10/P10008.

[4] Guerrero, M. et al. 2017. Adaptive community detection in complex networks using genetic algorithms. *Neurocomputing*. 266, (2017), 101–113. DOI:https://doi.org/10.1016/j.neucom.2017.05.029.

[5] Liu, D. et al. 2013. Genetic Algorithm with a Local Search Strategy for Discovering Communities in Complex Networks. *International Journal of Computational Intelligence Systems*. 6, 2 (2013), 354–369. DOI:https://doi.org/10.1080/18756891.2013.773175.

[6] Liu, H. et al. 2016. Genetic algorithm optimizing modularity for community detection in complex networks. *Chinese Control Conference, CCC*. 2016–Augus, 1 (2016), 1252–1256. DOI:https://doi.org/10.1109/ChiCC.2016.7553259.

[7] Newman, M.E.J. and Girvan, M. 2004. Finding and evaluating community structure in networks. *Physical Review E*. 69, 2 (Feb. 2004), 026113. DOI:https://doi.org/10.1103/PhysRevE.69.026113.

[8] Rao, A. et al. 2018. Efficient Reduced-Bias Genetic Algorithm (ERBGA) for Generic Community Detection Objectives. *MWAIS 2018 Proceedings*. 32 (May 2018).

[9] Said, A. et al. 2018. CC-GA: A clustering coefficient based genetic algorithm for detecting communities in social networks. *Applied Soft Computing Journal*. 63, (2018), 59–70. DOI:https://doi.org/10.1016/j.asoc.2017.11.014.

[10] Shi, C. et al. 2010. A Genetic Algorithm for Detecting Communities in Large-Scale Complex Networks. *Advances in Complex Systems*. 13, 01 (Feb. 2010), 3–17. DOI:https://doi.org/10.1142/S0219525910002463.

[11] Tasgin, M. et al. 2007. Community Detection in Complex Networks Using Genetic Algorithms. (Nov. 2007).